

**AUTOMATING THE TRAINING DEVELOPMENT PROCESS  
FOR MISSION FLIGHT OPERATIONS**

**Carol J. Scott**  
**Jet Propulsion Laboratory, California Institute of Technology**  
**4800 Oak Grove Drive M/S 230-101**  
**Pasadena, CA 91109**  
**(818) 393-0551**

**ABSTRACT**

Traditional methods of developing training do not effectively support the changing needs of operational users in a multimission environment. The Automated Training Development System (ATDS) Copyright © 1993, California Institute of Technology, provides advantages over conventional methods in quality, quantity, turnaround, database maintenance, and focus on individualized instruction. The Operations System Training Group at the Jet Propulsion Laboratory performed a six-month study to assess the potential of ATDS to automate curriculum development and to generate and maintain course materials. To begin the study, the group acquired readily available hardware and participated in a two-week training session to introduce the process.

ATDS is a building activity that combines training's traditional information-gathering with a hierarchical method for interleaving the elements. The program can be described fairly simply. A comprehensive list of candidate tasks determines the content of the database; from that database, selected critical tasks dictate which competencies of skill and knowledge to include in course material for the target audience. The training developer adds pertinent planning information about each task to the database, then ATDS generates a tailored set of instructional material, based on the specific set of selection criteria. Course material consistently leads students to a prescribed level of competency.

**INTRODUCTION**

The term "training" is open to interpretation. In its strictest sense, training (as opposed to education) can be described as a process enabling an individual to consistently produce a desired result. Individuals are trained to follow an established pattern of behaviors; they do not necessarily need to know what they are doing, why they are doing it, or what the result of their actions means in terms of an end product. In industry, a management goal is to get a product of consistent quality out the door in a timely manner. Computers are now widely used to automate labor-intensive operations. As complex software tools evolve to shave seconds off production time, training requirements become more intuitive in nature.

However, as commercial software has evolved, so has competition. To be successful in the marketplace, a program must be reliable and easy to learn. A significant investment in market research has produced software products with imbedded training tools for easier learning. In fact, learning ease has prompted the development of graphical user interfaces and other imbedded tools that nearly eliminate the need for formal training altogether. Complex processing mechanisms are nearly invisible to users of most commercial software, and command line entry has been virtually replaced with option selection via a mouse click. Users don't have to understand the process in order to work effectively with today's off-the-shelf programs.

A different production strategy is used in the development of custom, in-house software. With no marketing concerns and no competition, development time, effort, and funds are concentrated on functional design, enhancements as new ideas surface, and fixes as problems arise. At the Jet Propulsion Laboratory (JPL), custom mission operation software is developed, tested, and delivered to coincide with specific events. Many single-purpose programs are created to enable a limited number of users to support whatever unmanned spacecraft missions are currently active. Flexibility has a much higher priority than learning ease in mission operations software, so users are required to have great insight into the nature of the task in order to make effective on-line decisions while using the software.

As the name implies, multimission ground data systems software is being used by multiple spacecraft projects in differing stages of their mission life cycle. For example, the Voyager and Galileo projects are converting from their original systems to the Multimission Ground Data System, while Mars Observer launched using it; Magellan began life with an early version of the multimission system and decided to stick with that early version, rather than to evolve with the system as new versions were released. Cassini has not yet launched, but is planning its operational activities around the multimission system. Each project is made up of teams that focus on one or more functional activities; each team includes members that handle one or more elements of team concern. Each individual member has a unique perspective on software functionality and looks to training for guidance in meeting readiness goals.

Spacecraft operations personnel are often responsible for one-shot science opportunities that must be identified and pursued by dedicated, knowledgeable people working together as a well-rehearsed unit. A team may have some experienced "old-timers" from previous missions, but will always include new recruits who must begin training with the basics. Each team relies on its members to react intuitively to anomalous conditions, as well as on-going activities, so team members must understand the process, as well as the procedure.

JPL's Operations System Training Group designs and implements workstation training for mission operations personnel. Our resources for workstation training include a dedicated classroom, two real-time-access workstations running the current Multimission Ground Data System software, on-going system and configuration support (graphical user interfaces are beginning to appear for some tools and mature subsystems), two dedicated training developer/instructors, and outside consultation support from subject matter experts.

Since traditional training development methods generally rely on functional requirements to provide the impetus for generating courses and since procedural training is sequential in nature, it is often assumed that there exists one correct method, and that the procedures for that method can be performed by individuals with little or no related knowledge or experience. In fact, some commercial training seminars have concluded that even the trainer need not know anything about a subject to design an effective course. NOT SO for our training group! In fact, we must give the term "training" the broadest of definitions.

## **ROOM FOR PROCESS IMPROVEMENT**

As training developers, we are aware of training deficiencies, but lack personnel and resources to redesign our methods. Training is based on Instructional System Development (ISD) methods, but courses often lack consistency and structure. Our workstation training weaknesses have included a throughput bottleneck, cut-and-paste course maintenance, inefficient tracking and reporting methods, insufficient planning for future course development, and an on-going seat-of-the-pants approach to meeting customer needs on a "best efforts" basis. Requirements are difficult to define, especially during mission build-up when staffing deadlines are being met while job definitions are still fuzzy. Training materials are often based on software capability, rather than customer need.

A demonstration of the Automated Training Development System (ATDS) attracted our curiosity. Initial interest was monetarily based; ATDS was in-house technology that might be inexpensive to adapt for our implementation. The system was already in use for procedural training on a tactical military program, with documented success, and it seemed to have potential for our use in developing effective, maintainable training materials. ATDS, however, had not been tested for use with highly intuitive, computer-dependent data-analysis subject matter, but we were able to get funding for a six-month pilot study to assess the system's potential for use in our dynamic operational environment.

ATDS was originally developed to aid in the production of training materials for the JPL-managed All Source Analysis System project. System functionality is based on MIL-STD-1379D training specifications and conforms to current ISD training philosophies. We used ATDS on a Macintosh IIcx with a 19-inch monitor, 20-MB of RAM, an 80-MB hard disk, a 9600-baud modem, and an additional external storage device with removable 90-MB disks.

## LEARNING ATDS

Two weeks of training were held at the developer's site in Killeen, Texas. An existing beginning-workstation course in need of updating was used as a working project in learning the system, and an instructional task flow was developed prior to the training in order to get a head start on the course design process.

ATDS training was divided into two parts: design logic and hands-on practice. The program developer and database manager provided logical details about program design and functionality. The goal for the week was to gain a general understanding of the process. This understanding included traditional training development concepts, logical sequencing techniques for nesting competencies (individual skills and/or knowledge) within each task, and suggestions on how to determine the depth of detail necessary.

<b>ATDS Learning Process</b>	
<b><u>Logical Steps</u></b>	<b><u>ATDS Steps</u></b>
• list candidate tasks	• create tasks
• determine prerequisites	• enter prerequisite skills and knowledge
• determine enabling competencies	• develop skill and knowledge hierarchy
• describe each competency	• enter narratives
• determine user need	• survey analysis
• determine course content	• select critical tasks

*Figure 1. Automated Training Development System (ATDS) Learning Process.*

Figure 1 summarizes the ATDS learning process. For any given learning situation, there is a finite set of tasks that could be performed. This list of tasks will be referred to as "candidate tasks." The first step in creating a subject database is to define each task from the list of candidate tasks using ATDS. In order to perform each individual task, prerequisite skills or knowledge may be required prior to learning the task. These prerequisites may be entered once the task has been defined. Each task is made up of a set of competencies (skills and/or knowledge) that must be acquired during the learning process in order to complete the task. A Skill and Knowledge Hierarchy is subordinated to each task and may be nested up to six indenture levels. (Skills are usually supported by at least one knowledge and sometimes dependent upon lesser skills. See Figure 2.) Narratives describing each skill and knowledge in detail are then added to the database, along with other pertinent information about each task.

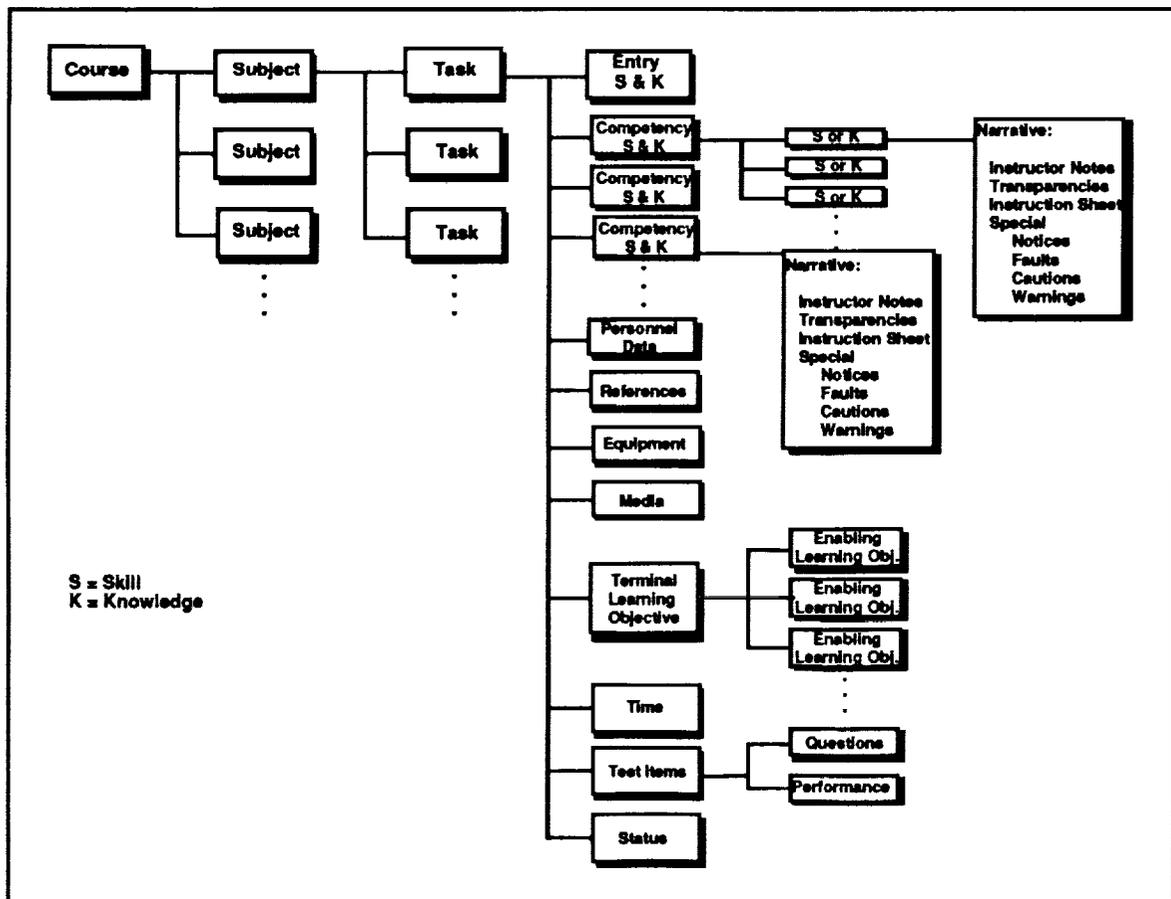


Figure 2. Database Hierarchy of the Automated Training Development System (ATDS).

Once the subject is thoroughly described within the database, the process of determining user requirements begins. A survey is taken to determine the task needs of the target audience. The results of the survey are fed into ATDS and a modeling process weighs the results to determine general criticality for training each task, based on survey consensus. The tasks that are selected for inclusion in the course material are called "critical tasks," since they are required by the target audience in order to complete a job.

During ATDS training, we recognized significant potential for supporting our current training development efforts. Drawbacks such as slow response time and numerous task changes were accepted as temporary in order to remain focused on the possibilities of ATDS. Many of our ideas for improving performance were incorporated into the program, and several more are being considered for future implementation.

### IMPLEMENTING ATDS

During the learning process, several issues surfaced as important goals needing closer attention in future training efforts:

- (1) Establish structured subject baselines. Instead of starting from scratch each time a new set of course material is needed, maintain and draw from a single-source all the material for a given subject, start to finish.
- (2) Build a solid foundation of skills and knowledge to the level of detail required to guarantee that complete mastery of a given subject is not lost with time and/or attrition.

- (3) Track development activities for future planning purposes. As new or enhanced subject matter is developed, estimates of development time for creating and populating databases and subsequently generating course materials should become predictable, based on growing experience.
- (4) Develop reporting mechanisms for relaying statistical training information to management.

In reviewing the contents of the database for the first time, something was missing from the overall task and competency subordination scheme; the sequences didn't flow correctly. The level of detail was then examined in the original course material and found to be inconsistent.

Meanwhile, a survey was prepared in order to exercise the critical task selection model for determining individual course content. The original basic workstation course was analyzed using the ATDS selection model to verify critical task selection activity with real survey results in order to fairly assess its value.

Figure 3 describes the content of the survey. The survey process had not yet been automated, so many hours were spent creating survey forms, writing introductory memos to participants and their managers, then distributing, and finally collecting the surveys. Our initial survey form was cumbersome and lengthy, and it missed the point about identifying tasks. Instead, it asked detailed questions about subordinate skills that were basically irrelevant to achieving the goal of task selection.

**Difficulty, Importance, and Frequency (DIF) Survey**

Task (Identify the task)

Question #1: Is the task Difficult?  
Select one of the following answers:  
a. Very difficult  
b. Somewhat difficult  
c. Not difficult

Question #2: Is the task Important?  
Select one of the following answers:  
a. Important  
b. Not important

Question #3: Is the task performed Frequently?  
Select one of the following answers:  
a. Very frequently  
b. Somewhat frequently  
c. Not frequently

*Figure 3. Critical Task Selection Survey.*

Analyzing the results seemed impossible without computer assistance. At least forty responses were recommended for adequate analysis, but in its existing state, there was no way to increment values within the model, and there were 27 possible values to consider. (See Figure 4.) The response to each question was weighted according to the response to the previous question. Questions with incomplete responses were thrown out altogether. The analysis process seemed endless, and the first five tasks all revealed the same thing--training was not required.

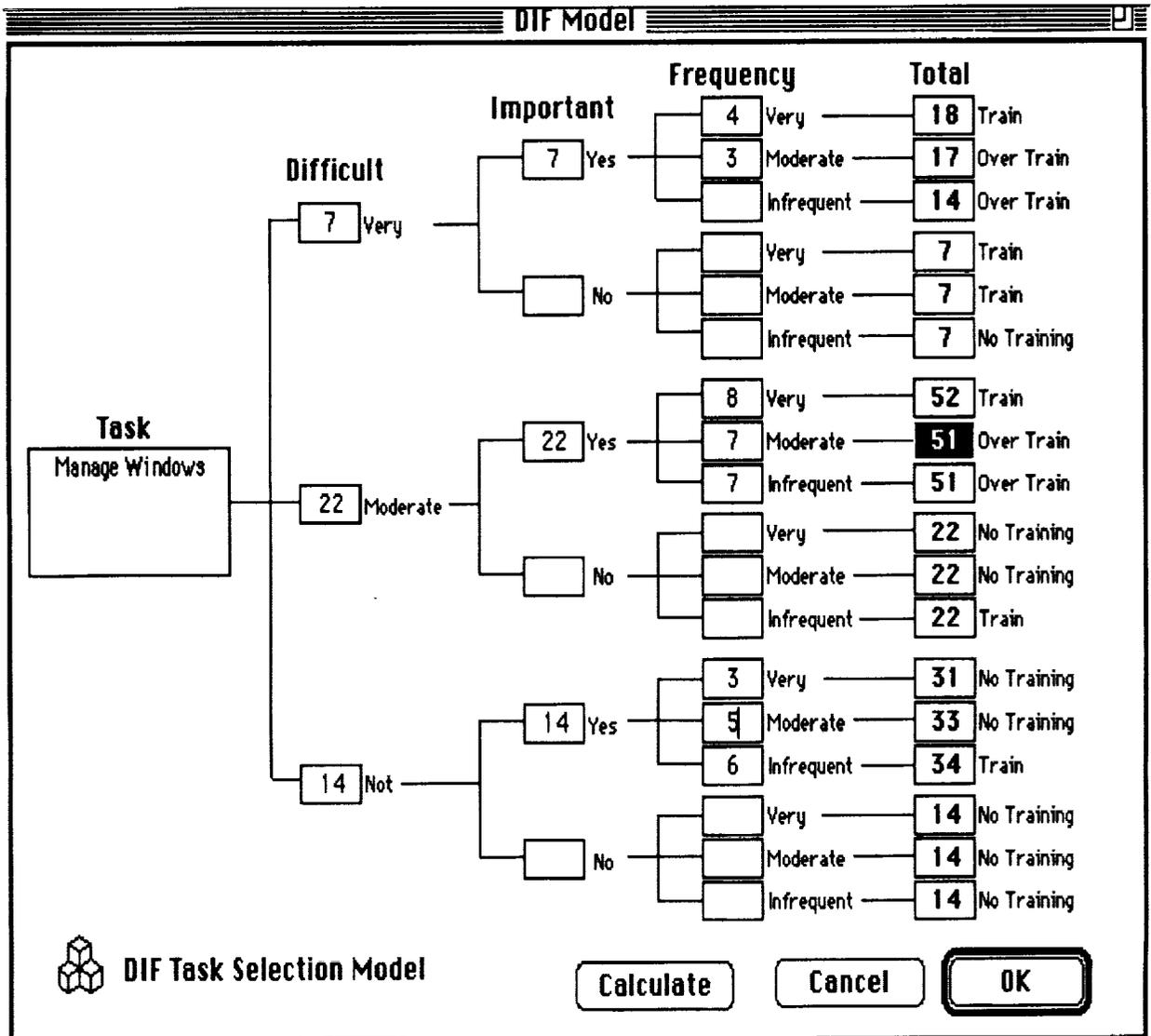


Figure 4. ATDS Difficulty, Importance, and Frequency Selection Model.

It took only a few calculations to realize that the original course content did not adequately meet student needs. A random sampling of task results revealed that the selected tasks were not critical tasks at all; instead, the original course was presumed to contain only critical tasks, and the ATDS development effort simply forced the expected results, rather than using ATDS as a vehicle to redesign the content of the course.

The importance of the survey became very clear. The training task list included an incomplete set of candidate tasks. It was the intent of the survey to filter out the unnecessary tasks and provide the training developer with a list of critical tasks. It became apparent that very little of the actual course content was included in the original course materials. Instead, a lot of "off-the-cuff" information had been directed at individuals in response to their questions, but had not been written down anywhere. The survey analysis provided solid insight into a potential single-point-of-failure in training.

ATDS could now provide us with a repository for narrative descriptions of each skill and knowledge (see Figure 5) that had previously gone undocumented.

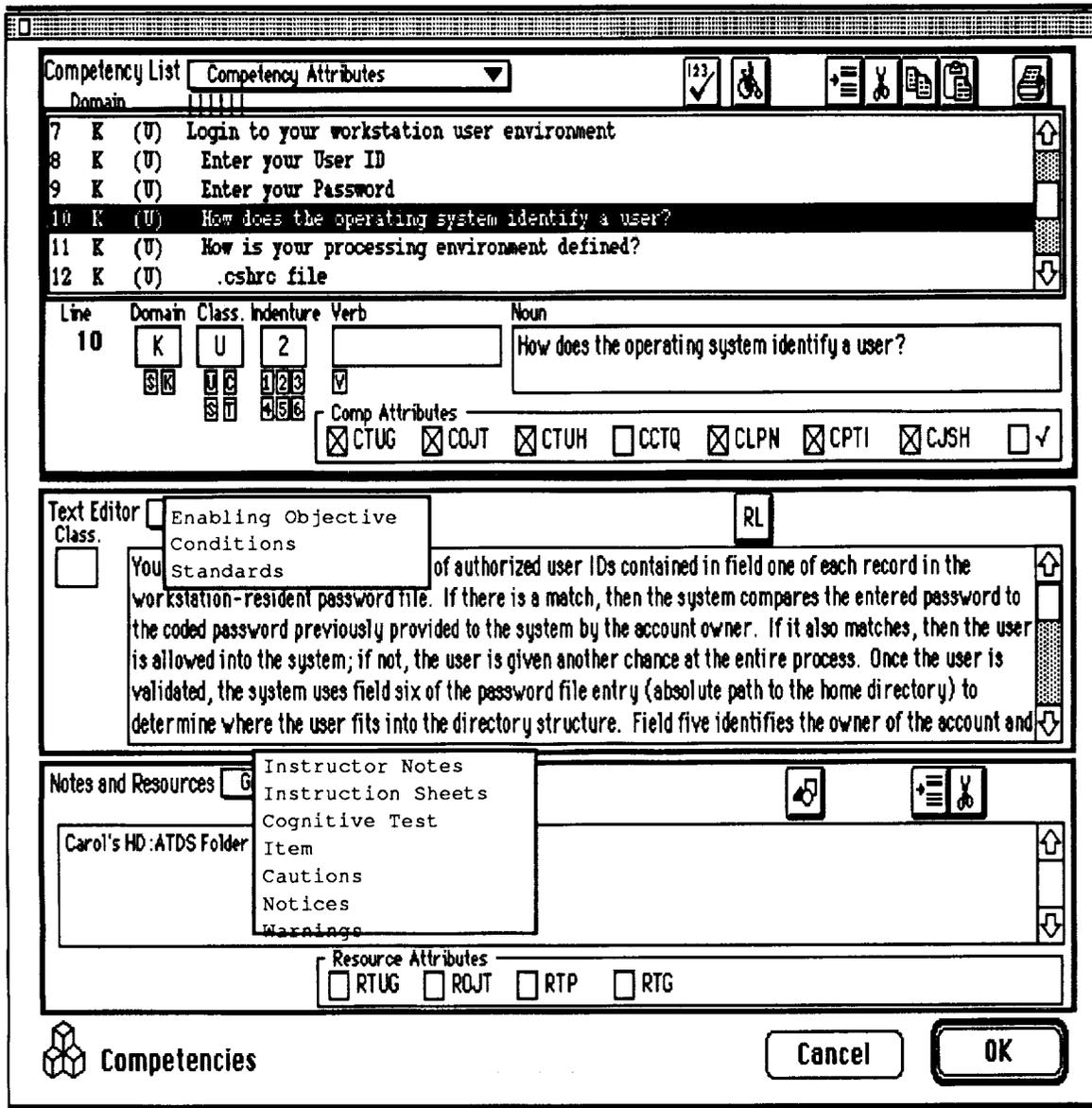


Figure 5. ATDS Skill and Knowledge Hierarchy Data Input Screen.

## FINDINGS

ATDS has limitations, but its potential for grasping the global nature of subject matter has had great impact on our perspective of the knowledge gathering activity.

The program is tedious to learn. It is not as easy to grasp as it appears. One of the problems with structured course development is that new information relevant to the subject matter floods the mind faster than it can be incorporated into the database, and constant juggling between competency levels is required. The reload delay between competencies causes frustration when thoughts dissolve before they can be captured.

Once all the tasks and their competencies are entered, along with their narratives and other supporting information, database maintenance and course generation are fairly simple. Developers can generate a course for a given set of users based on their needs, and can also generate any number of tailored courses that provide training by way of classroom instruction or self-guided workbooks for on-the-job training.

In its final form, the material is organized and easy to use. We found that ATDS product formats are consistent in quality and quantity, based upon database contents, so serious attention to input detail is essential.

During a dry-run session with partially completed materials, the problems encountered were the result of inexperience in dealing with such structured materials. Once "ad-libbing" is eliminated and explicit instructions followed, the materials were easy to use. In an emergency, substitute trainers should be able to implement course materials with very little notice.

### **Potential**

Considering the many aspects of our operations training development, ATDS has potential for improving the way we gather and maintain course material by preserving the global perspective of the subject. ATDS is a springboard for creative ideas. It is an effective organizing and building tool for thoughts and processes, and its basic functionality could be adapted to other uses, such as building and maintaining requirements.

ATDS requires a significant commitment of time and effort on the part of training developers to collect all the pieces of information and design learning sequences to best use each piece. The reward is a smoothly choreographed training presentation.

### **Not for Everyone**

ATDS can be a tedious learning experience and those who have never agonized over the design of a training module may have difficulty sticking with the process.

Those who have never designed a learning sequence, and probably never will again, need to have someone design the training for them. They could probably learn to maintain the material, once a baseline is established, but for a small training task it might not be worth the time.

Those who have no training background, but need to design training materials (for instance, to teach a new subordinate how to perform the job), will probably want to work with a consultant. Again, a solid subject baseline is the goal, although the training developer might feel comfortable with routine material maintenance once the baseline is created.

For those whose training requirements change significantly and frequently, possibly never to be repeated, ATDS is probably not a solution. The value of ATDS lies in its consistency and maintainability over time.

### **Who Will Benefit?**

ATDS will be most cost effective over time. Initial training database development activity will probably be a lengthy process; its value will be realized in quality, flexibility, and maintainability. However, the basics of design are not limited to trainers, and with some labeling changes within the program, this development process could take on many design formats.

Long-range planners take a goal or an idea and break it down into basic milestones or components that build upon each other to form a whole. Like training developers, they start with a basic set of tasks or criteria that drives their design process and defines its boundaries. ATDS helps maintain focus and records the entire process to whatever depth of detail is desired.

Unlike planners, strategists work in the realm of possibilities. Many options may be available, and their job is to trace each to its end and determine which strategy will work best. Again, they are building something, only in multiples. ATDS may be helpful here because it allows all the details to be recorded to any desired level of detail. In this case, the mathematical model might be useful in gathering opinions and assessing political or financial support.

Anyone who has ever had to maintain documentation, presentations, or procedures over time will appreciate the capabilities of ATDS. Of course training developers must maintain all three simultaneously.

## CONCLUSION

ATDS is not an authoring system, nor does it interface directly with the student at this point in time. It has, however, ignited a new interest in course development that focuses on a single-source information retention system, individualized course tailoring, and easy maintenance. The program is still under development, but our preliminary study supports its continued use.

Many of our Operations System Training Group development activities will benefit from conversion to ATDS. It would enhance our ability to quickly respond to our customers' needs. In fact, our products that support new or modified software programs should be developed and maintained for release to coincide with the software deliveries. Updated training material could be a part of any software delivery, provided good communication existed between software development, training development, test, documentation, and configuration control personnel. (A significant motivator to enhance communication between these groups would be to tie project acceptance of the software to the availability of training and documentation.)

Another area that should benefit from ATDS is training throughput. Trainer/mentors might evolve from the training automation process. Once a course has been tailored to the needs of a team or group, it may be possible to enlist a representative member of that team, at the project's request, to oversee self-guided learning within the operational environment. Full training support and on-going training material maintenance and development would, of course, be required. Resident trainer/mentors might address a concern commonly voiced from operations management that their people can't be away from their jobs, and certainly not in groups, for training activities.

---

The research described in this paper is being carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.



**Session H4: MODELING IN SUPPORT OF OPERATIONS  
AND ANTHROPOMETRY**

---

**Session Chair: Ms. Barbara Woolford**

